2Г

УДК 004.424.2, 004.624

В.В. Ганченко, А.А. Дудкин, А.В. Инютин
Объединенный институт проблем информатики НАН Беларуси, г. Минск
Беларусь, 220012, г. Минск, ул. Сурганова, 6
{ganchenko, avin}@lsi.bas-net.by, doudkin@newman.bas-net.by

Конвертация топологических данных с использованием параллельных вычислений

V.V. Ganchenko, A.A. Doudkin, A.V. Inyutin
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Belarus, 220012, c. Minsk, Surganova st., 6
{ganchenko, avin}@lsi.bas-net.by , doudkin@newman.bas-net.by

Conversion of VLSI Layout Data Using Parallel Computing

В.В. Ганченко, А.А. Дудкін, А.В. Інютін
Об’єднаний інститут проблем інформатики НАН Білорусі, м. Мінськ
Білорусь, 220012, м. Мінськ, вул. Сурганова, 6
{ganchenko, avin}@lsi.bas-net.by, doudkin@newman.bas-net.by

Конвертація топологічних даних з використанням паралельних обчислень

В статье рассмотрены алгоритмы конвертации топологических данных из форматов САПР в формат Т29 (формат установок автоматического контроля топологических структур на фотошаблонах производства УП «КБТЭМ-ОМО»), который напрямую не поддерживается современными САПР. Особенность таких данных – большой объем данных и, следовательно, время конвертации. Использование параллельных методов вычислений на базе ПЭВМ с многоядерным процессором и технологии OpenMP позволяет значительно ускорить время обработки данных.
Ключевые слова: конвертации данных, параллельные вычисления, САПР, фотошаблоны.

In this paper, we propose algorithms and software for parallel implementation of conversion of VLSI Layout Data from CAD format Gerber and MEBES into an internal representation and then into format of automatic mask inspection system T29. The software is developed on basis of OpenMP technology to work on PC with 4-core processor. It is shown that using parallel computing speeds up a conversion process. In the future we are planning to develop similar tools for data conversion from CIF and DFX formats into the internal format, and further into format GDS-II. These tools allow developing a program complex of topological data processing to work with automatic mask inspection system made by R&D Company “KBTEM-OMO” of “Planar” Corporation.
Key Words: data conversion, parallel computing, VLSI, routing mask.

У статті розглянуті алгоритми конвертації топологічних даних з форматів САПР у формат Т29 (формат установок автоматичного контролю топологічних структур на фотошаблонах виробництва УП «КБТЕМ-ОМО»), який прямо не підтримується сучасними САПР. Особливість таких даних – великий обсяг даних і, відповідно, час конвертації. Використання паралельних методів обчислень на базі ПЕОМ з багатоядерним процесором і технології OpenMP дозволяє значно прискорити час обробки даних.
Ключові слова: конвертації даних, паралельні обчислення, САПР, фотошаблони.

Введение

Информатизация и компьютеризация современного общества требуют непрерывной разработки и совершенствования микроэлектронной цифровой техники. При этом

постоянно возрастает сложность ее аппаратных компонентов и топологий интегральных микросхем – количество топологических элементов может превышать сотни миллионов. Топология СБИС на выходе специализированных САПР не может напрямую быть передана на вход оборудования производства СБИС, т.к. генераторы изображений и установки автоматического контроля оригиналов топологии на фотошаблонах имеют собственное представление данных.

Конвертация данных между различными САПР широко применяется в практике, например для организации сквозного проектирования с использованием нескольких различных САПР или обмена данными между такими системами и внешними приложениями [1-7]. Зачастую при этом используются механизмы ускорения обработки или обмена между различными подсистемами [8]. Доступность многоядерных ПЭВМ на рынке делает выгодным использование параллельных методов обработки данных при конвертации, в том числе и для оборудования для производства СБИС. То есть актуальной в практическом плане *задачей* является ускорение конвертации больших объемов данных.

Целью данной работы является разработка алгоритмов и программных средств быстрого преобразования и анализа топологических данных из распространенных форматов САПР в данные специализированного оборудования производства СБИС. Формат данных такого оборудования разработан для высокой производительности контрольных установок, но напрямую не поддерживается распространенными на сегодня САПР. При этом высокая производительность установок автоматического контроля топологических структур на фотошаблонах производства УП «КБТЭМ-ОМО» является важным преимуществом на международном рынке оборудования производства СБИС. Разрабатываемое оборудование производства СБИС ориентировано на технологии производства 0,13 мкм и менее при сложности 10^8 – 10^9 топологических элементов. Но использование существующих программных средств для конвертации топологических данных может занимать больше времени, чем сама проверка. Так, например, контроль пластины размером 157×157 мм проводится за 20 минут, а конвертация данных для нее на ПЭМВ занимает больше 1 часа. Такое время конвертации является неприемлемым и обуславливает *актуальность* предлагаемого программного комплекса.

Главными зарубежными аналогами предлагаемого ПК ПТД являются пакеты CATS фирмы Synopsys (Mountain View, California, USA) и L-EDIT фирмы Tanner EDA (Monrovia, California, USA) [9,10]. Данные пакеты позволяют преобразовывать комплекс проектных данных в машинные инструкции для электронно-лучевых и лазерных генераторов, используемых для создания изображений (фотошаблонов) и производства IC, MEMS, TFT-LCD, TFH, фотоники и биочипов.

Например, ядро пакета CATS выполняет следующие функции:

1) преобразование многоугольников из форматов разработки (CIF, DXF и GDS-II) в прямоугольники и трапеции файла генератора изображений или базы данных;

2) преобразование прямоугольников и повернутых прямоугольников в формат данных оптического генератора;

3) масштабирование или калибровочные операции над набором данных.

Стоимость комплекта ПО, необходимого для конвертации топологических данных, составляет более 100 тыс. $ на 1 рабочее место.

Пакет L-EDIT фирмы Tanner EDA представляет собой топологический редактор, предназначенный для синтеза топологии в автоматическом или интерактивном режимах, проверки правил проектирования, экстракции схемы проекта из топологии, сохранения топологии проекта в виде файлов стандартных форматов (CIF, DXF, GDS-II).

Целью предлагаемой разработки является формирование из файлов во входном формате Gerber или MEBES файла выходного формата T29 (формат описания топологии фотошаблона установок контроля) за максимально короткое время.

Задача решается в три этапа:

1) из файла либо файлов во входном формате формируются файлы в формате внутреннего представления данных. В этом формате описание топологии представляет набор контуров с неотрицательными значениями координат узлов. Все операции масштабирования, зеркального отображения, поворота изображения, аппроксимации и т.д., заданные во входном формате, должны быть выполнены;

2) из файлов в формате внутреннего представления данных формируются файлы промежуточного формата. Для этого контура должны быть разбиты на примитивы. Примитив – это контур, у которого координаты узлов по оси X от минимальной до максимальной координаты сверху и снизу монотонно неубывающие. Кроме этого примитивы должны быть разрезаны по границам полос;

3) из файлов промежуточного формата формируется файл выходного формата T29.

Ускорения работы программного комплекса достигается организацией вычислений на базе многоядерного процессора и использовании технологии OpenMP [11-14].

Методы распараллеливания вычислительных потоков для работы на многоядерной архитектуре ПЭВМ

Наиболее распространенные методы программирования параллельных приложений:

– метод с использованием Threads, который не требует дополнительных библиотек, т.е. для использования этого варианта достаточно имеющихся возможностей ОС. Обычно используется для сокрытия от пользователя различных операций, чтобы не допустить остановку отрисовки GUI в момент ожидания операции. Также достоинством является разделение адресного пространства и принадлежность потоков одному процессу. Поэтому все передачи данных между потоками выполняются максимально быстро. Чаще всего достаточно передать указатель. Синхронизация потоков не требует множества ресурсов и системных вызовов, которые являются длительными операциями с переключением контекста;

– более простым методом для распараллеливания уже существующего кода является стандарт OpenMP, т.к. значительная часть работы по распараллеливанию и синхронизации здесь переложена на компилятор и его библиотеки. Недостатки – требуется специальный компилятор, низкий уровень параллельности, необходимость следования навязываемой парадигме;

– библиотека передачи сообщений MPI представляет собой собрание функций на C/C++ (или подпрограмм в Фортране), облегчающих обмен данными и синхронизацию задач между процессами параллельной программы с распределенной памятью. MPI чаще всего применяют для написания параллельных программ на больших кластерах. Преимущества применения MPI – это высокая масштабируемость, высокий уровень параллельности и отличная портабельность кода. Основные минусы – сложности при программировании, относительно высокие затраты на синхронизацию и обмен данными.

Поскольку, согласно техническому заданию, разработка ПК ПТД будет осуществляться на основе последовательного кода, то для работы выбран стандарт OpenMP.

Стандарт OpenMP был разработан в 1997 г. как API, ориентированный на написание портируемых многопоточных приложений. Сначала он был основан на языке

Fortran, но позднее включил в себя и C/C++. Последняя версия OpenMP — 2.0; ее полностью поддерживает Visual C++ 2005 и более новые среды разработки Microsoft.

Для активации реализованных в компиляторе средств OpenMP служит появившийся в Visual C++ 2005 параметр компилятора /openmp. (возможна активация директивы OpenMP на страницах свойств проекта, через выбор в Configuration Properties, C/C++, Language и изменения значения свойства OpenMP Support). Встретив параметр /openmp, компилятор определяет символ _OPENMP, с помощью которого можно определить, включены ли средства OpenMP.

OpenMP связывается с приложениями через библиотеку импорта vcomp.lib. Соответствующая библиотека периода выполнения называется vcomp.dll. Отладочные версии библиотек импорта и периода выполнения (vcompd.lib и vcompd.dll соответственно) поддерживают дополнительные сообщения об ошибках, генерируемых при некоторых недопустимых операциях.

Для реализации параллельных вычислений выбран открытый стандарт для распараллеливания программ OpenMP, который позволяет достаточно легко и прозрачно реализовывать параллельные вычисления.

Сравнение производительности различных архитектур

Для сравнения производительности вычислений на CPU и GPU было реализовано два программных модуля. Основным различием этих модулей является обрабатываемый тип данных. Тестирование производительности CPU осуществлялось в двух режимах: режим, использующий только одно ядро, и режим, использующий четыре ядра процессора (реализовано с использованием OpenMP).

Тестирование производительности осуществлялось путем выполнения несложных математических операций над двумя массивами данных большой размерности (52 444 777 элементов в обоих случаях). Этими операциями являются:

$$\sqrt{mas1(i)^2 + mas2(i)^2} \tag{1}$$

для модуля, использующего тип данных float и

$$mas1(i)^2 + mas2(i)^2 \tag{2}$$

для модуля, использующего тип данных int.

Вычислительные модули тестировались на следующей программно-аппаратной платформе:

Процессор – AMD Phenom II X4 955 3.21 ГГц.

Видеокарта – NVIDIA GeForce GTS 250.

Объем ОЗУ, МБ – 4 Гбайт.

ОС – Windows XP SP3.

Производительность оценивалась путем измерения времени, затраченного модулем на выполнение одной задачи (вычисление выражений (1) и (2)) различными способами: одно ядро CPU, четыре ядра CPU (OpenMP) и GPU.

Результаты тестирования показаны на рис. 1.

Из рис. 1 следует, что использование параллельных вычислений при обработке больших объемов данных всегда позволяет сократить временные затраты. Однако способ распараллеливания следует выбирать исходя из условий решаемой задачи. Так, использование GPU будет оправданным только в случае большого количества однотипных и достаточно сложных вычислений над данными, представленными в виде

чисел с плавающей запятой. В случае же более простых вычислений либо при работе с целочисленными данными выгоднее использовать распараллеливание вычислений на отдельные ядра процессора, с помощью спецификации OpenMP.

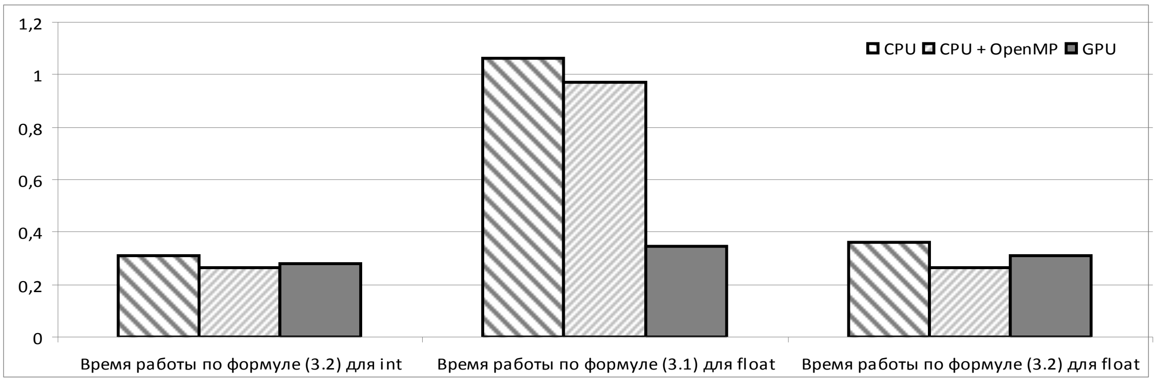


Рисунок 1 – Временные диаграммы результатов тестирования

Параллельная реализация преобразования топологических данных

Формат данных MEBES

Процесс конвертации исходных контуров, представленных в формате MEBES, состоит из нескольких этапов (рис. 2):

1) чтение исходного контура;

2) преобразование данных об объекте топологии;

3) сохранение полученного описания объекта топологии и в файл формата внутреннего представления.

Формат данных Gerber

Процесс конвертации исходных контуров, представленных в формате Gerber, состоит из нескольких этапов:

1) чтение команд из файла формата Gerber в список;

2) выборка команд из списка прочитанных команд плоттеру и их выполнение;

3) выполнение действий по смешению, повороту и масштабированию изображения;

4) сохранение результата выполнения в файл внутреннего формата.

Алгоритм разбора файла в формате Gerber RS-274X можно описать как автомат лексического разбора текстового файла, на вход которого посимвольно подаются исходные данные. Файл формата состоит из многих блоков данных (команд), содержащих параметры и коды. Каждый блок данных ограничивается символом конца блока. Каждый блок данных может содержать один или несколько параметров или кодов. Файл RS-274X может содержать расширенные данные (расширенные команды или X-данные) и стандартные коды RS-274D (D, G, N, M-коды).

Алгоритм разбора файла данных можно представить следующими (обобщенными) шагами (схема алгоритма приведена на рис. 3):

001. Проверка доступа к лог-файлу (файлу сообщений);

002. Инициализация состояния преобразования;

003. Доступ к файлам данных (.gbr + .int);

004. Инициализация состояния «плоттера» (только на первом вхождении);

005. Проверка начала команды (если расширенная команда, то разбор расширенной команды (на шаг 006), иначе – разбор простой команды (на шаг 010));

006. Разбор части расширенной команды (варианты):

007. Поиск завершения части расширенной команды (если нашли – то на шаг 008, иначе – ошибка);

008. Выполнение части расширенной команды (если команда только меняет состояние плоттера, то смена состояния производится в процессе разбора команды (см. шаг 006), а не по ее завершению (по завершению только констатируется факт обработки));

009. Поиск окончания расширенной команды (если найден конец расширенной команды, то на шаг 005, иначе на шаг 006);

010. Разбор простой команды (варианты):

011. Поиск окончания простой команды (если нашли – то на шаг 012, иначе – ошибка);

012. Выполнение найденной команды (если команда только меняет состояние плоттера, то смена состояния производится в процессе разбора команды (см. шаг 010), а не по ее завершению (по завершению только констатируется факт обработки));

013. Переход на шаг 005;

014. Действия по смешению, повороту и масштабированию изображения (не поддерживаются внутренним форматом);

В данном алгоритме при любой выявленной ошибке разбора или формирования внутреннего формата происходит аварийное завершение программы (не указано в обобщенной пошаговой схеме из-за сложности представления).

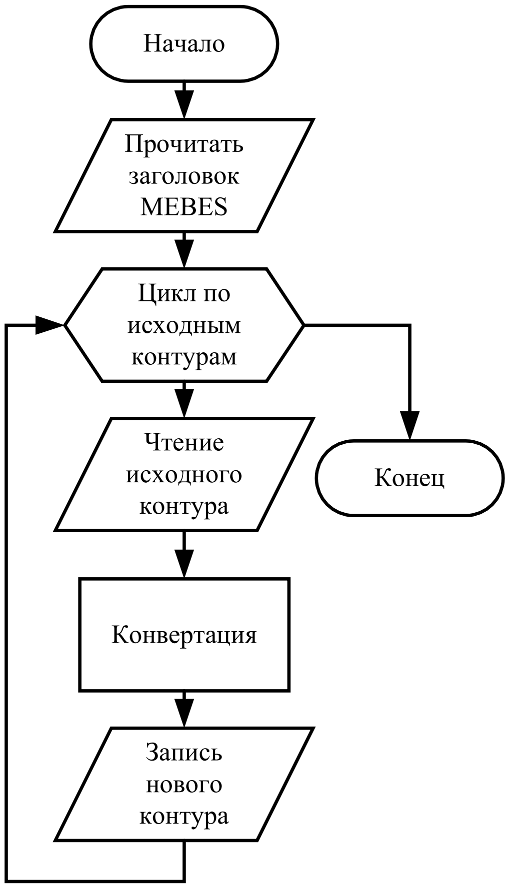


Рисунок 2 –Последовательная реализация алгоритма преобразования формата MEBES во внутреннее представление

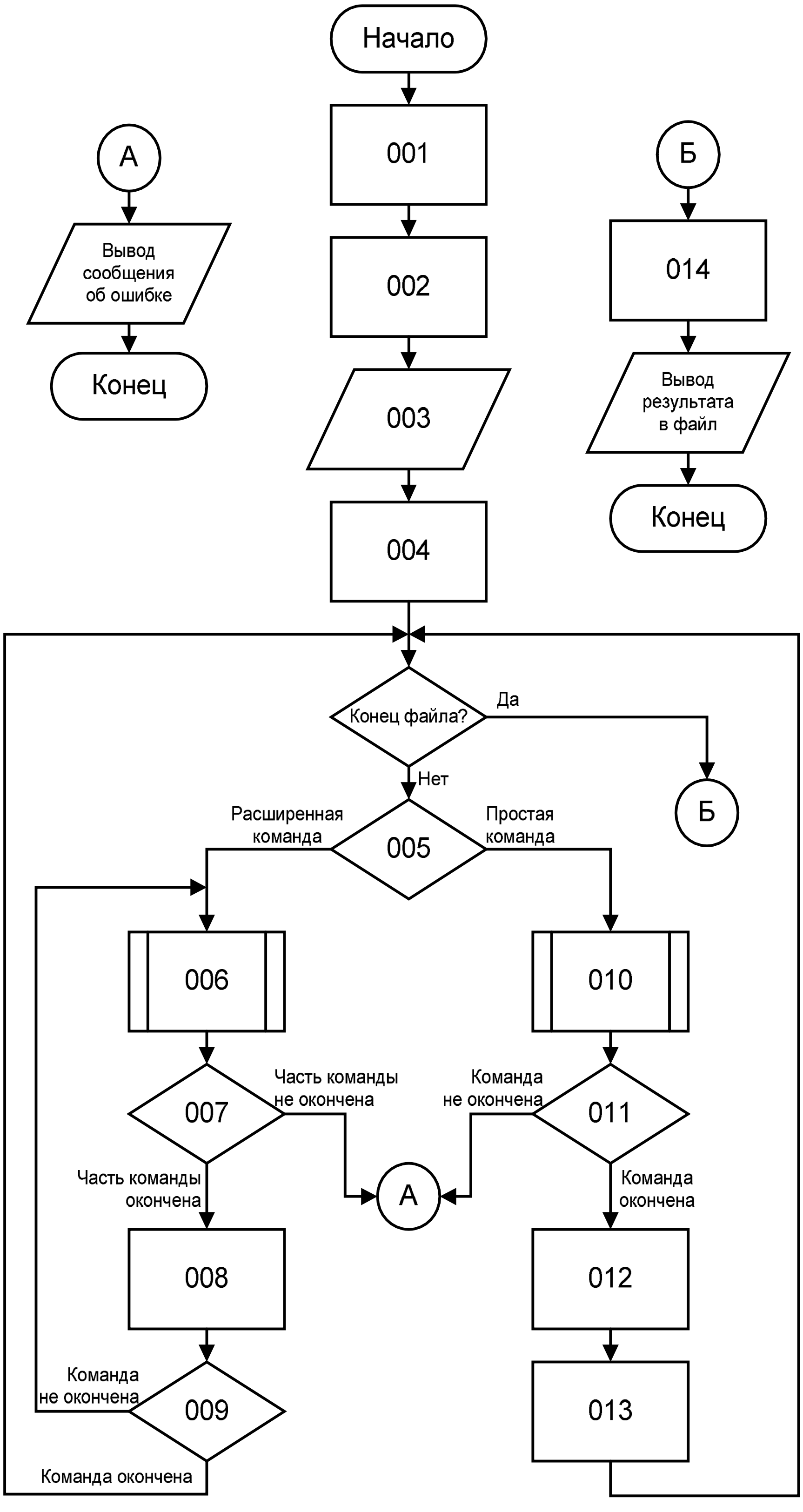


Рисунок 3 – Последовательная реализация алгоритма преобразования формата Gerber во внутреннее представление

Параллельная реализация преобразования данных в форматах CAD во внутреннее представление

При работе с данными в формате MEBES и Gerber, процесс преобразования можно представить в виде трех логически и функционально разделенных блоков, каждый из которых может выполняться параллельно с другими.

Для обеспечения требуемого порядка контуров при параллельной работе модуля преобразования формата предлагается использовать структуру данных «очередь» для передачи данных между этапами преобразования (рис. 4).

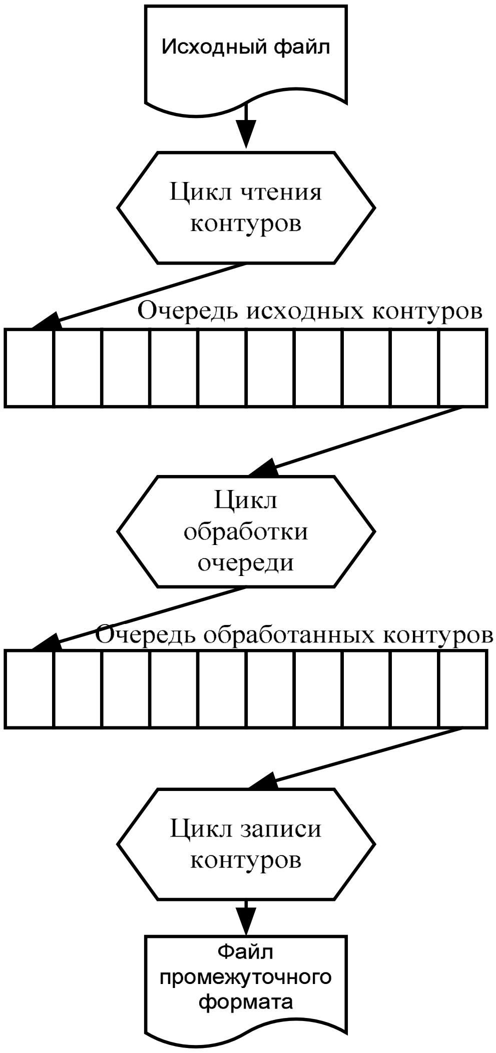


Рисунок 4 – Параллельная реализация алгоритма преобразования данных в форматах MEBES и Gerber во внутреннее представление

При такой схеме передачи данных работа модуля конвертации данных в формате MEBES осуществляется следующим образом:

1) чтение заголовка MEBES-файла;

2) [Поток 1] Циклическое вычитывание контуров из исходного файла и помещение их в очередь исходных контуров;

3) [Поток 2] Циклическое преобразование вычитываемых из очереди исходных контуров данных и сохранение результата обработки в очереди обработанных контуров;

4) [Поток 3] Циклическое вычитывание данных из очереди обработанных контуров и их сохранение в результирующий файл внутреннего представления.

Таким образом, работа параллельного алгоритма преобразования контуров может быть представлена в виде схемы, приведенной на рис. 5.

Работа модуля конвертации данных в формате Gerber может быть описана схемой, представленной на рис. 5, и осуществляется следующим образом:

1) открыть файл формата Gerber;

2) [Поток 1] циклическое чтение команд из Gerber-файла, ее разбор и запись в очередь на обработку (передача команды плоттеру на выполнение);

3) [Поток 2] вычитывание команды плоттеру для обработки и запись полученного результата в очередь на сохранение в файл внутреннего формата;

4) [Поток 3] циклическое вычитывание данных из очереди результатов выполнения команд и сохранение данных в выходной файл внутреннего формата.

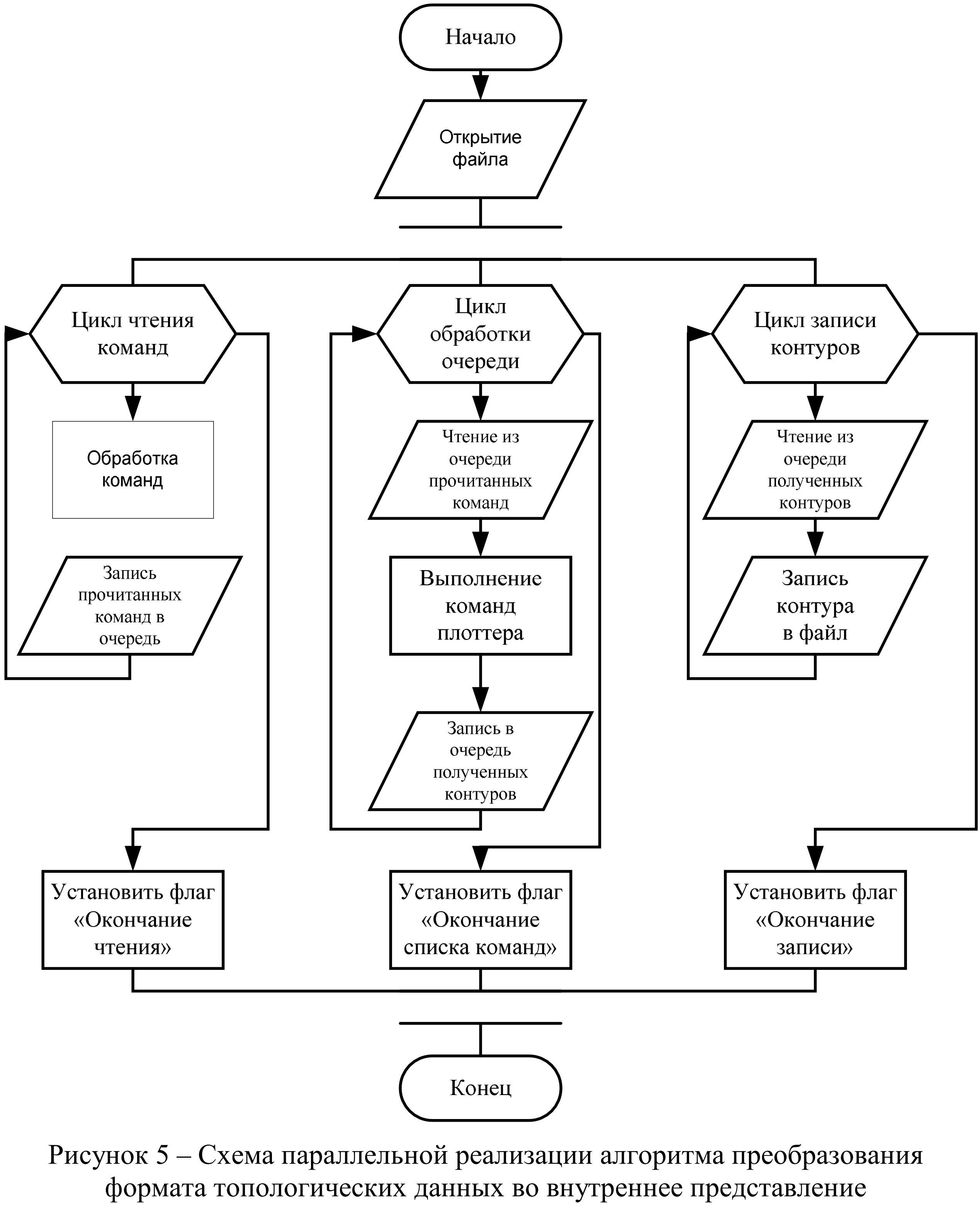


Рисунок 5 – Схема параллельной реализации алгоритма преобразования формата топологических данных во внутреннее представление

Работа по такой схеме требует выполнения некоторых условий:

1) требуется обеспечение синхронизации доступа к очередям;

2) введение дополнительных флагов, сигнализирующих о состоянии других потоков обработки (например, поток преобразования должен быть оповещен о том, что все исходные данные вычитаны, и цикл преобразования может быть завершен после опустошения очереди исходных контуров).

Формат установок автоматического контроля Т29

Для увеличения скорости работы полной процедуры конвертации топологических данных из формата внутреннего представления в промежуточный формат используется разбиение контуров на примитивы и дальнейшее преобразование в формат установок автоматического контроля Т29 реализовано одновременное вычисление в нескольких потоках. В качестве основы для параллельных вычислений была выбрана спецификация OpenMP [1], [9]. Такой выбор обусловлен структурой исходных программных модулей, а также используемой аппаратной платформой (персональный компьютер с многоядерным процессором и большим объемом оперативной памяти).

Данные программные модули используют конвертор, последовательная реализация которого разработана в УП «КБТЭМ-ОМО». Структура последовательной реализации программной системы приведена на рис. 6 а) (приведена только часть схемы, реализующая процедуру конвертации из формата внутреннего представления в формат Т29).

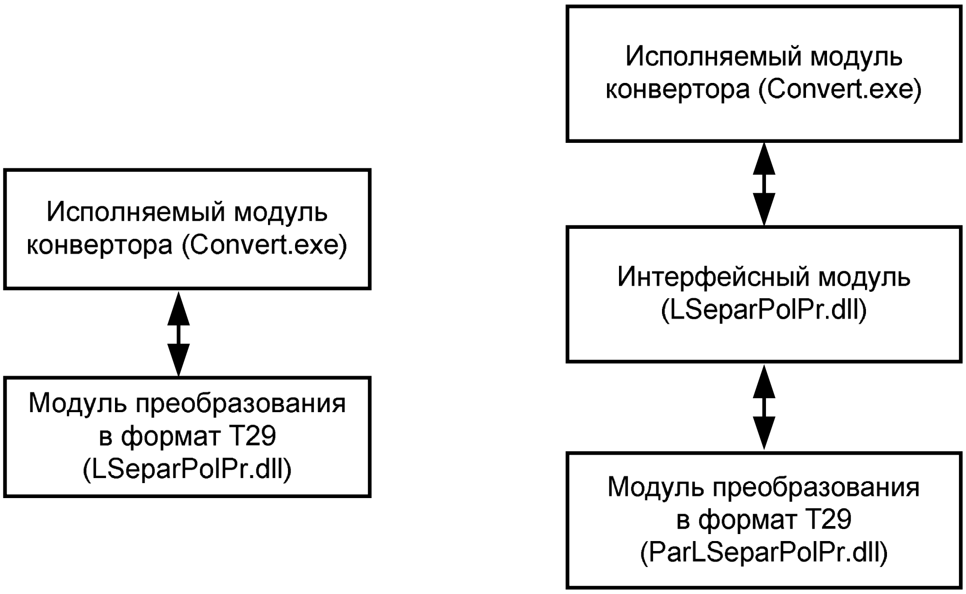


Рисунок 6 – Структурная схема а) исходной
и б) переработанной программной системы

Особенности реализации программной системы:

1) использование большого количества глобальных переменных;

2) использование среды разработки Borland C++ Builder 5, компилятор которой не поддерживает спецификацию OpenMP.

Данные особенности не позволяют выполнить распараллеливание вычислений с использованием структуры, приведенной на рис. 6, и технологии OpenMP. Поэтому вводится дополнительный уровень, который инкапсулирует вычисления в отдельной динамически загружаемой библиотеке. При этом исходная библиотека LSeparPolPr.dll используется лишь для осуществления взаимодействия с исполняемым модулем Convert.exe. Новая схема приведена на рис. 6 б).

В новой схеме интерфейсный модуль LSeparPolPr.dll осуществляет следующие задачи:

– инициализация переменных;

– подготовка данных к выполнению конвертации;

– загрузка модуля параллельного разбиения контуров на примитивы (ParLSeparPolPr.dll) и функции run_all() из данной динамически загружаемой библиотеки;

– запуск вычислений;

– выполняет постобработку (закрытие файловых дескрипторов, очистка памяти).

Библиотека ParLSeparPolPr.dll предназначена только для выполнения вычислений. Она экспортирует только три функции: ParallelPrepFileSeparConturs(), SetParameters() и run_all(), которые предназначены для загрузки данных из интерфейсного модуля и вычислений. Также модифицирована для работы с локальными переменными функция ConturProcessing(). Код новых и модифицированных функций реализован на языке программирования C++ в среде разработки Microsoft Visual Studio 2008.

Тестирование программных средств конвертации топологических данных

Для оценки сокращения временных затрат было проведено тестирование, в котором засекалось время конвертации для исходной версии модуля преобразования, новой версии модуля (использующей стандарт OpenMP) с одним задействованным вычислительным потоком и новой версии модуля с тремя задействованными вычислительными потоками. Тестирование осуществлялось на предоставленных УП «КБТЭМ-ОМО» файлах:

– топологии E5189ARR5.00 и файл пакетной обработки em5189test5090.jb (формат MEBES, размером 47 МБ: файл).

– ipc441_1.gbr, ipc441_2.gbr (формат Gerber, размер 1 Mb, время засекалось для 10 запусков последовательной и параллельной версий).

Результаты замеров временных затрат на преобразование из формата MEBES и Gerber в формат внутреннего представления приведены в табл. 1.

Таблица 1 – Сравнение времени работы различных реализаций конвертации данных

Формат	Время конвертации для модуля, с реализацией:		
	Последовательная	OpenMP и один поток	OpenMP и три потока
MEBES	1,517	1,593	1,304
Gerber	0,621	0,709	0,452

Из результатов тестирования можно заметить, что при использовании параллельной версией модуля только одного ядра и вычислительного потока время конвертации возрастает, что связано с затратами на распараллеливание, но при использовании трех потоков на трех ядрах время преобразования сокращается. Относительно невысокие результаты ускорения вычислений связаны с тем, что основными затратами при работе данного модуля (как последовательной, так и параллельной версий) являются затраты на чтение и запись данных на файловую систему.

Тестирование программных средств конвертации в формат T29 осуществлялось на файлах топологических данных во внутреннем формате, полученных преобразованием файла 112_01.gds. Исходный GDS файл содержит 903 контура, которые во внутреннем формате представляют собой 2030 примитивов.

В табл. 2 приведены средние значения времени работы конвертера на ПЭВМ с 4-ядерным процессором.

Таблица 2 – Результаты времени работы конвертера

Количество вычислительных потоков	Время работы
1	1,544
2	0,889
3	0,686
4	0,561

Из табл. 2 видно, что для 4-ядерной ПЭВМ выгодно использование нескольких вычислительных потоков.

Заключение

В данной работе предложены алгоритмы и программные средства параллельной реализации преобразования топологических данных в формате Gerber и пакетного режима MEBES во внутреннее представление и далее в формат установок автоматического контроля Т29. Программные средства разработаны на базе технологии OpenMP для работы на ПЭВМ с 4-ядерным процессором. Показано, что использование параллельных вычислений ускоряет процесс конвертации. В дальнейшем планируется разработать аналогичные средства для конвертации данных из форматов CIF и DFX во внутренний формат и из него – в формат GDS-II, что позволит создать программный комплекс подготовки топологических данных для работы с установками автоматического контроля производства УП «КБТЭМ-ОМО».

Литература

1. Wang W. Research on model data exchange method between different structure CAD systems / W. Wang, Zh. Hu, Ch. Liu // Proc. of International Conference on Multimedia Technology (ICMT), Hangzhou, China, 26 – 28 July 2011. – P. 3626-3629.
2. Standardized data exchange of CAD models with design intent / J. Kim [et al.] // Computer-Aided Design. – 2008. – V. 7(40). – P. 760-777.
3. Using Procedure Recovery Approach to Exchange Feature-based Data among Heterogeneous CAD Systems / X. Li [et al.] // Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009) April 22 – 24, 2009, Santiago, Chile. – P. 716-721.
4. Qi Z. Interdisciplinary Data Exchange in the Development of Assembly Systems / Zh. Qi, C. Schafer, P. Klemm // Proc. of the 4th IEEE International Conference on Industrial Informatics INDIN06, Singapore, 2006, 16 – 18 Aug. – P. 542-547.
5. Li A. An data exchange methodology of CAD and virtual assembly system / A. Li, X. Shen // Proc. of the IEEE 17Th International Conference on Industrial Engineering and Engineering Management (IE&EM), 29 – 31 Oct. 2010, Xiamen, China. – P. 307-310
6. Pajarre E. G2L: system for converting low-level geometrical designs to a higher level representation / E. Pajarre, T. Ritoniemi, H. Tenhunen // Proc. of the Euro ASIC '91, 27 – 31 May 1991, Paris, France. – P. 366-371.
7. Research on CAD Data Format Conversion for Transport Infrastructure Information / H. Wang [et al.] // Advanced Materials Research. – 2011. – Vol. 305. – P. 239-242.
8. An improved data communication mechanism for a SOC hardware/software co-emulation environment / A.W. Ruan [et al.] // Proc. of International Conference on Communications, Circuits and Systems ICCCAS 2009, Milpitas, California, USA, 23 – 25 July 2009. – P. 1029-1032.
9. CATS Synopsys [Электронный ресурс] – 2012. – Режим доступа : http://www.synopsys.com/tools/manufacturing/masksynthesis/cats/pages/default.aspx. – Дата доступа: 02.07.2012.
10. Tanner EDA Software Tools - L-Edit Layout for Analog IC & Mixed Signal Design [Электронный ресурс] – 2012. – Режим доступа : http://tannereda.com/l-edit-pro. – Дата доступа: 02.07.2012.
11. The OpenMP® API specification for parallel programming [Электронный ресурс] – 2011. – Режим доступа : http://openmp.org/wp/ . – Дата доступа: 19.09.2011.
12. Антонов А.С. Параллельное программирование с использованием технологии OpenMP : учебное пособие / Антонов А.С. – М. : Изд-во МГУ, 2009. – 77 с.
13. Боресков А.В. Основы работы с технологией CUDA / А.В. Боресков, А.А. Харламов. – М. : Изд-во ДМК-Пресс, 2010. – 232 с.
14. OpenCL Programming Guide / A. Munshi [et al]. – Addison-Wesley Professional, 2011. – 648 p.

Literatura

1. Wang, W. Research on model data exchange method between different structure CAD systems / W. Wang, Zh. Hu, Ch. Liu // Proc. of International Conference on Multimedia Technology (ICMT), Hangzhou, China, 26-28 July 2011. – P. 3626 – 3629.

2. Standardized data exchange of CAD models with design intent / J. Kim [et al.] // Computer-Aided Design. – 2008. – V. 7(40). – P. 760 – 777.
3. Using Procedure Recovery Approach to Exchange Feature-based Data among Heterogeneous CAD Systems / X. Li [et al.] // Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009) April 22-24, 2009, Santiago, Chile. – P. 716–721.
4. Qi, Z. Interdisciplinary Data Exchange in the Development of Assembly Systems / Zh. Qi, C. Schafer, P. Klemm // Proc. of the 4th IEEE International Conference on Industrial Informatics INDIN06, Singapore, 2006, 16-18 Aug. – P. 542 – 547.
5. Li, A. An data exchange methodology of CAD and virtual assembly system / A. Li, X. Shen // Proc. of the IEEE 17Th International Conference on Industrial Engineering and Engineering Management (IE&EM), 29-31 Oct. 2010, Xiamen, China. – P. 307 – 310
6. Pajarre, E. G2L: system for converting low-level geometrical designs to a higher level representation / E. Pajarre, T. Ritoniemi, H. Tenhunen // Proc. of the Euro ASIC '91, 27-31 May 1991, Paris, France. – P. 366–371.
7. Research on CAD Data Format Conversion for Transport Infrastructure Information / H. Wang [et al.] // Advanced Materials Research. – 2011. – Vol. 305. – P. 239-242.
8. An improved data communication mechanism for a SOC hardware/software co-emulation environment / A.W. Ruan [et al.] // Proc. of International Conference on Communications, Circuits and Systems ICCCAS 2009, Milpitas, California, USA, 23-25 July 2009. – P. 1029 – 1032.
9. CATS Synopsys [Jelektronnyj resurs] – 2012. – Rezhim dostupa: http://www.synopsys.com/tools/manufacturing/masksynthesis/cats/pages/default.aspx.
10. Tanner EDA Software Tools - L-Edit Layout for Analog IC & Mixed Signal Design [Jelektronnyj resurs] – 2012. – Rezhim dostupa: http://tannereda.com/l-edit-pro.
11. Kompleksy optiko-mehanicheskogo oborudovanija dlja bezdefektnogo izgotovlenija fotoshablonov urovnja 0,35mkm i 90 nm / S.M. Avakov [i dr.] //Fotonika. – 2007. – #6. – S.35–39.
12. The OpenMP® API specification for parallel programming [Jelektronnyj resurs] – 2011. – Rezhim dostupa: http://openmp.org/wp/ . – Data dostupa: 19.09.2011.
13. Antonov, A.S. Parallel'noe programmirovanie s ispol'zovaniem tehnologii OpenMP: Uchebnoe posobie. M.: Izd-vo MGU. 2009. 77 s.
14. Boreskov, A.V. Osnovy raboty s tehnologiej CUDA. Izd-vo: DMK-Press. 2010. 232 s.
15. OpenCL Programming Guide / A. Munshi [et al]. – Addison-Wesley Professional, 2011. – 648 p.
16. Widmann, D. Technology of Integrated Circuits /D. Widmann, H. Mader, H. Friedrich – Berlin: Springer, 2000. – 360 p.
17. Opisanie formata Gerber RS-274X / Barco company [Jelektronnyj resurs] – Rezhim dostupa: http://www.barco.com/ets/data/rs274xc.pdf.

RESUME

V.V. Ganchenko, A.A. Doudkin, A.V. Inyutin

Conversion of VLSI Layout Data Using Parallel Computing

In this paper, we propose algorithms and software for parallel implementation of conversion of VLSI Layout Data from CAD format Gerber and MEBES into an internal representation and then into format of automatic mask inspection system T29. The software is developed on basis of OpenMP technology to work on PC with 4-core processor. It is shown that using parallel computing speeds up a conversion process. In the future we are planning to develop similar tools for data conversion from CIF and DFX formats into the internal format, and further into format GDS-II. These tools allow developing a program complex of topological data processing to work with automatic mask inspection system made by R&D Company "KBTEM-OMO" of "Planar" Corporation.

Статья поступила в редакцию 03.07.2012.